

REMARKS

Claims 1-6, 8-20 and 22-30 are currently pending in the subject application and are presently under consideration. Claims 1, 22 and 26 have been amended as shown on pp. 2-6 of the Reply. Claim 20 has been canceled.

Applicant's representative thanks the Examiner for the courtesies extended during the teleconference of August 20, 2007.

Since the amended limitations merely emphasize subject matter as originally claimed, these limitations should already have been considered during an initial search in connection with the subject application. Pursuant to MPEP §714.13, applicant's representative submits that the amendments to these claims "only requires a cursory review by the Examiner" and thus, entry and consideration thereof is respectfully requested.

Favorable reconsideration of the subject patent application is respectfully requested in view of the comments and amendments herein.

I. Rejection of Claims 1-18 and 20 Under 35 U.S.C. §101

Claims 1-18 and 20 stand rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. Independent claim 1 has been amended herein to clearly illustrate that elements within such claims are components associated with a computer processor. In particular, claim 1 as amended is directed towards a computer-based event handling system, recorded on a computer-readable medium and capable of execution by a computer, comprising *a computer processor for executing the following software components*, a framework component, a synchronization component and a configuration component. (Support for these amendments can be found on pg. 6, lines 7-20). Accordingly, this claim includes functional descriptive material within a computer processor, thereby rendering it structurally and functionally interrelated to the computer processor and is therefore directed to statutory subject matter. Claim 20 has been canceled. Accordingly, this rejection should be withdrawn with regard to claims 1-18 and 20.

II. Rejection of Claims 1-18 and 20 Under 35 U.S.C. §112, second paragraph

Claims 1-18 and 20 stand rejected under 35 U.S.C. §112, second paragraph as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Claim 1 has been amended to correct any deficiencies related

to this rejection and claim 20 has been canceled. As such, the rejection is moot and should be withdrawn.

III. Rejection of Claims 1-6, 8-11, 18-20 and 22-30 Under 35 U.S.C. §103(a)

Claims 1-6, 8-11, 18-20 and 22-30 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Allavaru *et al.* (US Patent 6,839,748) in view of Avery *et al.* (US 2005/0005259). It is respectfully requested that this rejection should be withdrawn for at least the following reasons. Allavaru *et al.* and Avery *et al.*, individually or in combination, do not teach or suggest each and every element as set forth in the subject claims.

To reject claims in an application under §103, an examiner must show an un rebutted *prima facie* case of obviousness. A *prima facie* case of obviousness is established by a showing of three basic criteria. First, there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. Second, there must be a reasonable expectation of success. Finally, the prior art reference (or references when combined) must teach or suggest all the claim limitations. *See* MPEP §706.02(j). The teaching or suggestion to make the claimed combination and the reasonable expectation of success must both be found in the prior art and not based on applicant's disclosure. *See In re Vaeck*, 947 F.2d 488, 20 USPQ2d 1438 (Fed. Cir. 1991).

The claimed subject matter relates to synchronization of threads in a multi-threaded event-driven environment. The claimed subject matter employs an object structured system environment in which classes of objects raise events. The object system determines what synchronization is required for respective events based on a pre-categorization of the objects and associated instances. In particular, independent claim 1 recites an event handling system, comprising: *a framework component that supplies classes of objects that can raise events; a synchronization component that controls in part synchronization of access to data based on categorization of at least one of objects and instances defined by the framework, and provides automatic serialization for events raised by the classes of objects; and a configuration component that enables the client component to disable or enable automated serialization and synchronization; wherein the framework component allows a device driver to allocate additional memory, such that the device driver stores its working data in this allocated object*

extension memory and relies on the framework component for concurrency management and serialization. Allavarpu *et al.* and Avery *et al.*, individually or in combination, do not expressly or inherently disclose the aforementioned novel aspects of applicant's claimed subject matter as recited in the subject claims.

Allavarpu *et al.* discloses a system for a synchronous task scheduler. The synchronous task scheduler is used with a CORBA Gateway between CORBA-based client manager applications and an enterprise manager. The CORBA Gateway includes an Event Gateway which manages events from managed objects, and a Request Gateway which manages requests and responses of managed objects. The Event Gateway and the Request Gateway is designed as multi-threaded systems. A thread pool may be used to increase efficiency and performance of the CORBA Gateway. The thread pool provides multiple threads delivering events to multiple consumers, such that if one consumer is slow in processing events, other consumers should not be adversely affected. (See col. 6, line 41-col. 7, line 54).

In contrast, applicant's claimed subject matter discloses an event handling system. The system includes a framework component having a synchronization component that provides automatic serialization for events that are raised by one or more objects. The framework, synchronization component, and/or associated object automatically manage or serialize the events in order that a client component can process other desired tasks at hand (*e.g.*, process driver-specific tasks rather than processing timing issues). The events are managed to allow one or more aspects of the events to occur in a one-at-a-time manner and in accordance with a serialized process.

Furthermore, the system includes a configuration component (*e.g.*, Application Programming Interface (API)) to enable the client component to disable and/or enable automated serialization and synchronization. The framework provides events to a device driver through a series of callback functions registered by the device driver. To assist a driver writer or code designer, the framework offers configurable serialization that allows simpler techniques to achieve a desired level of performance, while still allowing a more complex driver to turn off the serialization and achieve behavior with a possible increase in performance. This is opposed to a synchronous or "blocking" model in which a read event handler submits the read, and then waits within the read event handler for an event signaled by the device when the read is complete.

Applicant's claimed framework is structured around state full objects that allow the

device driver to register events, and provide API's. These objects are represented by the handle passed to the device driver. The framework allows the device driver to allocate additional memory for its own use *via* the objects handle. The object oriented nature of the framework encourages a device driver to store its working data in this allocated "object extension", or "object context" memory that is related to the object for access within event handler functions. The overall structure of the driver's data within the object's context memory along with its event driven, non-blocking nature is useful for providing a model in which the device driver can rely on the framework for the bulk of its concurrency management and serialization. (See pg. 6, line 21-pg. 9, line 11).

Allavarpu *et al.* merely discloses the use of a synchronous task scheduler that preserves the chronology of messages sent to the client manager. When messages are delivered serially to the client manager application, such messages are scheduled by the synchronous task scheduler. The synchronous task scheduler maintains an internal message list, and delivers one message at a time from that internal list. (See col. 22, lines 5-45). Applicant's claimed system allows the device driver to allocate additional memory for its own use *via* the objects handle, wherein the overall structure of the driver's data within the object's context memory along with its event driven, non-blocking nature is useful for providing a model in which the device driver can rely on the framework for the bulk of its concurrency management and serialization. This is opposed to a synchronous or "blocking" model, as in Allavarpu *et al.*, in which a read event handler submits the read, and then waits within the read event handler for an event signaled by the device when the read is complete. Accordingly, Allavarpu *et al.* does not expressly or inherently disclose a system that provides ... *a configuration component that enables the client component to disable or enable automated serialization and synchronization; wherein the framework component allows a device driver to allocate additional memory, such that the device driver stores its working data in this allocated object extension memory and relies on the framework component for concurrency management and serialization.*

Avery *et al.* does not cure the deficiencies of Allavarpu *et al.* with respect to independent claims 1, 20, 22 and 26, Avery *et al.* discloses a system for communication and mapping of business objects between a mobile client device and a plurality of backend systems *via* a network. The system includes a mobile server and a mobile client device in data communication with the mobile server. The mobile server includes a process automation engine configured to

map business objects to the plurality of backend systems and including a plurality of mobile business processes, each mobile business process defining a flow and exchange of business objects between the mobile client device and at least one of the plurality of backend systems. (See pg. 1, paragraph [0007]).

Avery *et al.* does not disclose allowing the device driver to allocate additional memory for its own use *via* the objects handle and encouraging a device driver to store its working data in this allocated “object extension”, or “object context” memory that is related to the object for access within event handler functions. Thus, the overall structure of the driver’s data within the object’s context memory along with its event driven, non-blocking nature is useful for providing a model in which the device driver can rely on the framework for the bulk of its concurrency management and serialization. Avery *et al.* merely discloses a synchronization management module that interfaces with the business object framework for translation, marshalling, serializing and unmarshalling business objects. Accordingly, Avery *et al.* also does not expressly or inherently disclose a system that provides ... *a configuration component that enables the client component to disable or enable automated serialization and synchronization; wherein the framework component allows a device driver to allocate additional memory, such that the device driver stores its working data in this allocated object extension memory and relies on the framework component for concurrency management and serialization.*

In view of the aforementioned deficiencies of the cited art, it is respectfully submitted that this rejection be withdrawn with respect to independent claims 1, 20, 22 and 26 (and claims 2-6, 8-11, 18, 19, 23-25 and 27-30 which respectively depend there from).

IV. Rejection of Claims 12-17 Under 35 U.S.C. §103(a)

Claims 12-17 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Allavarpu *et al.* in view of Avery *et al.*, and further in view of Mandal *et al.* (US 2004/0117369). It is respectfully requested that this rejection should be withdrawn for at least the following reasons. Allavarpu *et al.*, Avery *et al.* and Mandal *et al.*, individually or in combination, do not teach or suggest each and every element as set forth in the subject claims. In particular, Mandal *et al.* does not make up for the aforementioned deficiencies of Allavarpu *et al.* and Avery *et al.* with respect to independent claim 1 (which claims 12-17 depend from). Thus, the subject

invention as recited in claims 12-17 is not obvious over the combination of Allavarpu *et al.*, Avery *et al.* and Mandal *et al.*

CONCLUSION

The present application is believed to be in condition for allowance in view of the above comments and amendments. A prompt action to such end is earnestly solicited.

In the event any fees are due in connection with this document, the Commissioner is authorized to charge those fees to Deposit Account No. 50-1063 [MSFTP490US].

Should the Examiner believe a telephone interview would be helpful to expedite favorable prosecution, the Examiner is invited to contact applicant's undersigned representative at the telephone number below.

Respectfully submitted,

AMIN, TUROCY & CALVIN, LLP

/Marisa J. Zink/

Marisa J. Zink

Reg. No. 48,064

AMIN, TUROCY & CALVIN, LLP
24TH Floor, National City Center
1900 E. 9TH Street
Cleveland, Ohio 44114
Telephone (216) 696-8730
Facsimile (216) 696-8731